

Package ‘CIS.DGLM’

January 20, 2025

Type Package

Title Covariates, Interaction, and Selection for DGLM

Version 0.1.0

Description An implementation of double generalized linear model (DGLM) building with variable selection procedures and handling of interaction terms and other complex situations. We also provide a method of handling convergence issues within the `dglm()` function. The package offers a simulation function for generating simulated data for testing purposes and utilizes the forward stepwise variable selection procedure in model-building. It also provides a new custom bootstrap function for mean and standard deviation estimation and functions for building crossplots and squareplots from a data set.

License GPL (>= 2)

Encoding UTF-8

Imports ggplot2, dglm, RColorBrewer, dplyr

RoxygenNote 7.1.2

NeedsCompilation no

Author Ann Stapleton [aut],
Yishi Wang [aut, cre],
Kaitlyn Hohmeier [aut],
Jordan Tanley [aut]

Maintainer Yishi Wang <wangy@uncw.edu>

Repository CRAN

Date/Publication 2022-01-17 08:32:52 UTC

Contents

bootstrap	2
draw.crossplots	3
draw.squareplots	4
forward.sel.dglm	6
forward.sel.dglm.real	6
mean_stress	7
sd.stress	8
simu.inter.dat.interboth	10

bootstrap	<i>Bootstrap</i>
-----------	------------------

Description

This function implements a custom bootstrapping procedure that utilizes bootstrapping to estimate mean and SD of stress between two environment states (A and B).

Usage

```
bootstrap(
  dataset,
  n.boot = 10^5,
  variables,
  stress_variable,
  alpha = 0.05,
  ran.seed = 12345
)
```

Arguments

dataset	Data set to be utilized.
n.boot	Number of bootstraps to perform. Defaults to 10 ⁵ .
variables	List of variables from mean and variance models in DGLM.
stress_variable	Name of the variable with the stress values.
alpha	Significance level by which to determine the confidence intervals for the bootstrap estimates. Defaults to 0.05, thus creating the 95 percent confidence intervals.
ran.seed	Random seed value for generating different random bootstrap samples.]

Value

Lists with confidence intervals for the bootstrap estimations for average stress in As and Bs of variables in mean model and confidence intervals for the bootstrap estimations of standard deviation of stress in As and Bs of variables in variance model.

Examples

```
test.data <- simu.inter.dat.interboth(n.rep = 3, n.obs.per.rep = 15, ran.seed = 1)
variables <- colnames(test.data[-1])
bootstrap(test.data, n.boot=100,variables, 'stress')
unlink(c('bootstrap mean A stress.txt','bootstrap mean B stress.txt',
'bootstrap sd A stress.txt', 'bootstrap sd B stress.txt'))
```

draw.crossplots	<i>Draw Crossplots</i>
-----------------	------------------------

Description

This function draws crossplots for As and Bs in each variable in the mean and variance models with the Mean Estimate vs Standard Deviation Estimate.

Usage

```
draw.crossplots(  
  fn.mean.A,  
  fn.mean.B,  
  fn.sd.A,  
  fn.sd.B,  
  fn.pe.mean,  
  fn.pe.sd,  
  variables,  
  ishybrid,  
  num.vars  
)
```

Arguments

fn.mean.A	Enter file name of file with confidence intervals of mean stress, environment level A data. Can be hybrid, inbred, or full data set. This file needs to be obtained from the bootstrap function, run with the desired data set.
fn.mean.B	Enter file name of file with confidence intervals of mean stress, environment level B data. Can be hybrid, inbred, or full data set. This file needs to be obtained from the bootstrap function, run with the desired data set.
fn.sd.A	Enter file name of file with confidence intervals of SD stress, environment level A data. Can be hybrid, inbred, or full data set. This file needs to be obtained from the bootstrap function, run with the desired data set.
fn.sd.B	Enter file name of file with confidence intervals of SD stress, environment level B data. Can be hybrid, inbred, or full data set. This file needs to be obtained from the bootstrap function, run with the desired data set.
fn.pe.mean	Enter file name of file with point estimates of mean for each gene (both A and B environment levels present). Can be hybrid, inbred, or full data set. This file needs to be obtained from the mean_stress function, run with the desired data set.
fn.pe.sd	Enter file name of file with point estimates of SD for each gene (both A and B environment levels present). Can be hybrid, inbred, or full data set. This file needs to be obtained from the sd.stress function, run with the desired data set.
variables	List of variables from mean and variance models. Mean variables need to be listed first, then variance variables.

ishybrid	Indicates the type of the data set being examined. You can use 'Hybrid', 'Inbred', "All", etc.
num.vars	Number of variables per model. Used to ascertain if a variable falls in the mean or the variance model.

Value

There is no return for this function; it prints crossplots for each of the variables listed in the parameter 'variables.'

Examples

```
test.data <- simu.inter.dat.interboth(n.rep = 3, n.obs.per.rep = 15, ran.seed = 1)
variables <- colnames(test.data[-1])
mean_stress(test.data, variables, 'stress')
sink();
sd.stress(test.data, variables, 'stress')
sink();
plot_vars <- c("loci_var.4", "loci_var.7.env_var.2", "loci_var.3",
"loci_var.5", "loci_var.8.env_var.2", "loci_var.4")
bootstrap(test.data, n.boot=100, variables, 'stress')
draw.crossplots('bootstrap mean A stress.txt', 'bootstrap mean B stress.txt',
'bootstrap sd A stress.txt', 'bootstrap sd B stress.txt', 'mean_stress.txt',
'sd_stress.txt', plot_vars, 'All', 3)
unlink(c('bootstrap mean A stress.txt', 'bootstrap mean B stress.txt',
'bootstrap sd A stress.txt', 'bootstrap sd B stress.txt',
'mean_stress.txt', 'sd_stress.txt'))
```

draw.squareplots

Draw Square Plots

Description

This function draws square plots for As and Bs in each variable in the mean and variance models with the Mean Estimate vs Standard Deviation Estimate

Usage

```
draw.squareplots(
  fn.mean.A,
  fn.mean.B,
  fn.sd.A,
  fn.sd.B,
  fn.pe.mean,
  fn.pe.sd,
  variables,
  ishybrid,
  num.vars
)
```

Arguments

fn.mean.A	file name of file with confidence intervals of mean stress, environment level A data. Can be either hybrid or inbred data.
fn.mean.B	file name of file with confidence intervals of mean stress, environment level B data. Can be either hybrid or inbred data.
fn.sd.A	file name of file with confidence intervals of SD stress, environment level A data. Can be either hybrid or inbred data.
fn.sd.B	file name of file with confidence intervals of SD stress, environment level B data. Can be either hybrid or inbred data.
fn.pe.mean	file name of file with point estimates of mean for each gene (both A and B environment levels present). Can be either hybrid or inbred data.
fn.pe.sd	file name of file with point estimates of SD for each gene (both A and B environment levels present). Can be either hybrid or inbred data.
variables	list of variables from mean and variance models. Mean vars needs to be listed first, then variance vars.
ishybrid	indicates the type of the data set being examined. Choose 'Hybrid' or 'Inbred' or "All".
num.vars	number of variables per model. Used to ascertain if a variable falls in the mean or the variance model.

Value

There is no return for this function; it prints square plots for each of the variables listed in the parameter 'variables'.

Examples

```
test.data <- simu.inter.dat.interboth(n.rep = 3, n.obs.per.rep = 15, ran.seed = 1)
variables <- colnames(test.data[-1])
mean_stress(test.data, variables, 'stress')
sink();
sd.stress(test.data, variables, 'stress')
sink();
bootstrap(test.data, n.boot=100, variables, 'stress')
plot_vars <- c("loci_var.4", "loci_var.7.env_var.2", "loci_var.3",
"loci_var.5", "loci_var.8.env_var.2", "loci_var.4")
draw.squareplots('bootstrap mean A stress.txt', 'bootstrap mean B stress.txt',
'bootstrap sd A stress.txt', 'bootstrap sd B stress.txt', 'mean_stress.txt',
'sd_stress.txt', plot_vars, 'All', 3)
unlink(c('bootstrap mean A stress.txt', 'bootstrap mean B stress.txt',
'bootstrap sd A stress.txt', 'bootstrap sd B stress.txt',
'mean_stress.txt', 'sd_stress.txt'))
```

forward.sel.dglm *Forward Stepwise Selection for Simulated Data*

Description

This function implements the forward stepwise variable selection procedure on the simulated data set generated in `simu.inter.dat.interboth`. In this function, we utilize a dummy value of "1" when initializing the model to avoid issues with a NULL value when adding variables to the model.

Usage

```
forward.sel.dglm(dat.ana.num12.df, ouput.name = "out1.txt", num.loop = 10)
```

Arguments

<code>dat.ana.num12.df</code>	A data set filled with data based on a simulation, per the procedure to generate it in the <code>simu.inter.dat.interboth</code> function.
<code>ouput.name</code>	The name of the output file to which the results are to be saved. Defaults to 'out1.txt'.
<code>num.loop</code>	The number of iterations that forward stepwise selection is performed (and hence how many variables will be in the final mean and variance models). Defaults to 10 loops.

Value

A list with mean and variance mean effects and p-values associated with the coefficients.

Examples

```
test.data <- simu.inter.dat.interboth(n.rep = 3, n.obs.per.rep = 15, ran.seed = 1)
forward.sel.dglm(test.data)
```

forward.sel.dglm.real *Forward Stepwise Selection for Real Data*

Description

This function implements the forward stepwise variable selection procedure on a real data set. It utilizes the `dglm` function from the `dglm` packages to build the model and helps to account for more complex situations such as convergence issues with `dglm` and interaction terms in the model. In this function, we utilize a dummy value of "1" when initializing the model to avoid issues with a NULL value when adding variables to the model.

Usage

```
forward.sel.dglm.real(
  dat.ana.num12.df,
  ouput.name = "out1.txt",
  num.loop = 10,
  typ.err = 0.05
)
```

Arguments

dat.ana.num12.df	The data set to be used to build the DGLM.
ouput.name	The name of the output file to which the results will be saved.
num.loop	The number of iterations that forward stepwise selection is performed (and hence how many variables will be in the final mean and variance models). Defaults to 10 iterations.
typ.err	Type 1 error. The default value is 0.05.

Value

A data frame with mean and variance mean effects and p-values associated with the coefficients for each loop. The function also produces a text file containing the model-building information at each stage of the loop (i.e. variables causing errors or warnings, the state of the model at each iteration, etc.).

Examples

```
library(dplyr)
test.data <- simu.inter.dat.interboth(n.rep = 3, n.obs.per.rep = 15, ran.seed = 1)
forward.sel.dglm.real(test.data)
unlink(c('out1.txt'))
```

mean_stress	<i>Mean Stress</i>
-------------	--------------------

Description

This function provides the mean stress among As and Bs, corresponding to different environment levels, for a list of variables.

Usage

```
mean_stress(
  dataset,
  var,
  stress_variable,
  output.name = "mean_stress.txt",
```

```

    use.output = TRUE,
    bin.levels = c(0, 1)
  )

```

Arguments

dataset	Data set to be utilized. For the purposes of this function, the binary values in each variable are considered to be -1 and 1.
var	A list of variables. If using variables from a DGLM, use the variables from the mean model (or, if trying to find intervals for use in the plotting functions draw.crossplots and draw.squareplots, use all variables in both mean and variance models).
stress_variable	Name of the variable with the stress values.
output.name	Name of the output file to which to save the outputs. Defaults to 'mean_stress.txt'.
use.output	A binary variable to indicate whether the output is automatically saved to an external text file. Defaults to TRUE. If FALSE, the output will not be saved to a file.
bin.levels	A list that provides the binary values utilized in the dataset. Defaults to c(0,1), indicating that 0 and 1 are used as the binary outcomes; can also be 1, -1. List the value for the "A" environment level first, then the value for the "B" environment level.

Value

Produces a data frame with three columns: var, AvgB, and AvgA. These provide the variable and its corresponding mean stress values for As and Bs, corresponding to different environment levels.

Examples

```

test.data <- simu.inter.dat.interboth(n.rep = 2, n.obs.per.rep = 15, ran.seed = 1)
variables <- colnames(test.data[-1])
mean_stress(test.data, variables, 'stress', use.output = FALSE)

```

sd.stress

Standard Deviation Stress

Description

This function provides the mean stress among As and Bs, corresponding to different environment levels, for a list of variables.

Usage

```
sd.stress(  
  dataset,  
  var,  
  stress_variable,  
  output.name = "sd_stress.txt",  
  use.output = TRUE,  
  bin.levels = c(0, 1)  
)
```

Arguments

dataset	Data set to be utilized. For the purposes of this function, the binary values in each variable are considered to be -1 and 1.
var	A list of variables. If using variables from a DGLM, use the variables from the variance model (or, if trying to find intervals for use in the plotting functions <code>draw.crossplots</code> and <code>draw.squareplots</code> , use all variables in both mean and variance models).
stress_variable	Name of the variable with the stress values.
output.name	Name of the output file to which to save the outputs. Defaults to 'sd_stress.txt'.
use.output	A binary variable to indicate whether the output is automatically saved to an external text file. Defaults to TRUE. If FALSE, the output will not be saved to a file.
bin.levels	A list that provides the binary values utilized in the dataset. Defaults to <code>c(0,1)</code> , indicating that 0 and 1 are used as the binary outcomes; can also be 1, -1. List the value for the "A" environment level first, then the value for the "B" environment level.

Value

Produces a data frame with three columns: `var`, `sd1`, and `sdneg1`. These provide the variable and its corresponding standard deviation of stress values for As and Bs, corresponding to different environment levels.

Examples

```
test.data <- simu.inter.dat.interboth(n.rep = 3, n.obs.per.rep = 15, ran.seed = 1)  
variables <- colnames(test.data[-1])  
sd.stress(test.data, variables, 'stress', use.output = FALSE)
```

```
simu.inter.dat.interboth
```

Simulated Data with DGLM model

Description

This function implements a simulation based on randomly-generated data following a known structure to create a double generalized linear model (DGLM). In particular, it supports the use of interaction terms in the DGLM.

Usage

```
simu.inter.dat.interboth(
  n.rep = 3,
  n.level.env = 2,
  n.obs.per.rep = 150,
  n.loci = 8,
  which.mean.loci = c(3:4),
  hypo.mean.para = c(1, 4),
  incept.mean = 36,
  which.var.loci = c(4:5),
  hypo.var.para = c(2, 5),
  incept.var = -2,
  which.env.inter.mean = 2,
  which.loci.inter.mean = 7,
  hypo.inter.para.mean = 2.5,
  which.env.inter.var = 2,
  which.loci.inter.var = 8,
  hypo.inter.para.var = 3.8,
  simu.prob.var = rep(0.5, n.loci),
  rc.val = 40,
  ran.seed = NULL
)
```

Arguments

<code>n.rep</code>	The number of repetitions of each environment level. Defaults to 3.
<code>n.level.env</code>	The number of environment variable levels. Defaults to 2.
<code>n.obs.per.rep</code>	A parameter to accommodate repetitions in the data set. Defaults to 150.
<code>n.loci</code>	The number of gene loci in data set. Defaults to 8.
<code>which.mean.loci</code>	A vector that specifies which gene loci are significant in the mean model. Defaults to the vector <code>c(3:4)</code> .
<code>hypo.mean.para</code>	A vector that contains the slopes of the gene locations (from <code>which.mean.loci</code>) in the mean model. Defaults to the vector <code>c(1,4)</code> .

<code>incept.mean</code>	The intercept for the mean model portion of the DGLM. Defaults to 36.
<code>which.var.loci</code>	A vector that specifies which gene loci are significant in the variance model. Defaults to <code>c(4:5)</code> .
<code>hypo.var.para</code>	A vector that contains the slopes of the gene locations (from <code>which.var.loci</code>) in the variance model. Defaults to <code>c(2,5)</code> .
<code>incept.var</code>	The intercept for the variance model portion of the DGLM. Defaults to -2.
<code>which.env.inter.mean</code>	Specifies which level of environment significantly interacts in the mean model. This is one of the parameters that supports model-building with interaction terms. Defaults to 2.
<code>which.loci.inter.mean</code>	Specifies which gene location interacts with the environment level in the mean model. This is one of the parameters that supports model-building with interaction terms. Defaults to 7.
<code>hypo.inter.para.mean</code>	The interaction effect between the specified environment level and the specified gene location in the mean model. This is another parameter that supports model-building with interaction terms. Defaults to 2.5.
<code>which.env.inter.var</code>	Specifies which level of environment significantly interacts in the variance model. This is another parameter that supports model-building with interaction terms. Defaults to 2.
<code>which.loci.inter.var</code>	Specifies which gene location interacts with the environment level in the variance model. This is another parameter that supports model-building with interaction terms. Defaults to 8.
<code>hypo.inter.para.var</code>	Interaction effect between the specified environment level and the specified gene location in the variance model. Defaults to 3.8.
<code>simu.prob.var</code>	The probability of each loci to be zero or one. Different loci may have different probability and can be adjusted as needed. Defaults to <code>rep(0.5,n.loci)</code> .
<code>rc.val</code>	A parameter used to control right-censored data. It sets an upper limit such that if an observation is above a certain range, that observation cannot be included in the data set. Default value is 40.
<code>ran.seed</code>	The random seed to be used. Defaults to NULL.

Value

The function returns a data frame containing the data built from the simulation. It provides data for stress values, an environment variable with two levels (0 and 1), and levels for each simulated gene variable.

Examples

```
test.data <- simu.inter.dat.interboth(n.rep = 3, n.obs.per.rep = 15, ran.seed = 1)
```

Index

`bootstrap`, [2](#)

`draw.crossplots`, [3](#)

`draw.squareplots`, [4](#)

`forward.sel.dglm`, [6](#)

`forward.sel.dglm.real`, [6](#)

`mean_stress`, [7](#)

`sd.stress`, [8](#)

`simu.inter.dat.interboth`, [10](#)